Online Action Detection

Roeland De Geest[1], Efstratios Gavves[2], Amir Ghodrati[1],
Zhenyang Li[2], Cees Snoek[2], Tinne Tuytelaars[1]

[1]KU Leuven - ESAT - PSI
[2]University of Amsterdam - QUVA

Abstract

In the computer vision problem of online action detection, the goal is to detect the start of an action in a video stream as soon as it happens. For instance, if a child is chasing a ball, an autonomous car should recognize what is going on and respond immediately. This is a very challenging problem. First, only partial actions are observed. Second, there is a large variability in negative data. Finally, in real world data, large within-class variability exists. This problem has been addressed before, but only to some extent.

First, we introduce a realistic dataset composed of 27 episodes from 6 popular TV series. The dataset spans over 16 hours of footage annotated with 30 action classes, totaling 6,231 action instances. Second, we analyze and compare various baseline methods with a newly-introduced evaluation protocol, showing this is a challenging problem for which none of the methods provides a good solution.

1 Introduction

In *online action detection*, unlike traditional action recognition and action detection (*e.g.*, [1, 4, 5, 9, 13, 17]), the goal is to detect an action as it happens and ideally even before the action is fully completed. Being able to detect an action at the time of the occurrence can be useful in many practical applications - think of a pro-active robot offering a helping hand; a surveillance camera raising an alarm not just after the facts but well in time to allow for intervention; a smart active camera system zooming in on the action scene and recording it from the optimal perspective; or an autonomous car stopping for a child chasing a ball (see Figure 1).

Current work on related problems assumes simplified setups [2, 6, 7, 11, 12, 15]. These are not representative for practical applications, where occurrences of any out of possibly many different action categories need to be detected in an online fashion, in (very) long video recordings with widely varying content. As we will show, this is a significantly more challenging task, to which the standard methods proposed in the literature provide only partial answers. Moreover, to date, no realistic benchmark dataset focusing on this problem has been released. To alleviate this problem, we introduce the *TVSeries* dataset, a new dataset consisting of 27 episodes of 6 popular TV series. The dataset is temporally annotated at the frame level w.r.t. 30 possible actions. Furthermore, metadata is added, containing extra information regarding the action occurrence.

Given a streaming video as input, the system should output, ideally in realtime, whether the action is currently taking place (or not). This requires detecting the ongoing action as accurately as possible, no matter what is the stage of the action. Since we focus on longer videos, this task requires in turn discriminating the action from a wide variety of negative data, including both background frames and irrelevant actions. For a TV series episode composed of 20 minutes of footage, a typical "standing up" action might not be appearing for more than 10 seconds in total. Only if a method can cope with this data imbalance and the large variability in the negative data, it will be of any practical use.

In summary, the challenges of real-world online action detection are the following. First, actions need to be detected as soon as possible, ideally after only part of the action has been observed. Second, actions need to be detected from among a wide variety of irrelevant negative data. Finally, we work with real world data, not artificially created for the purpose of action recognition. This results in large within-class variability.

Together with the TVSeries dataset, we also propose an evaluation protocol and we report initial results for a set of (state-of-the-art) baseline methods. As it turns out, detecting actions at the time of their occurrence in realistic settings, while keeping the number of false positives under control, is a much harder problem than one might conclude from results reported in the literature under more constrained settings. With this new dataset and evaluation protocol, we hope to encourage more researchers to look into the challenging yet very practical task of *online action detection*.

Figure 1: Illustration of an online action detection prediction.

2 Dataset

We collected the TVSeries dataset. The videos in this dataset depict realistic actions as they happen in real life. Our dataset is composed of professionally recorded videos: we annotated the first episodes of six recent TV series[1]. We select the number of episodes such that we have around 150 minutes of every series: almost 16 hours in total. We divide the episodes over a training, validation and testing set. We define 30 actions. We found 6,231 instances; every action has at least 50 instances. Actions are annotated manually: only temporally, not spatially.

Atypical	Does the actor perform the action in a way humans would call 'atypical'?
Multiple persons	Are there multiple persons during the action?
Small/background	Is the action very small or in the background?
Side viewpoint	Is the action recorded from the side?
Frontal viewpoint	Is the action from a frontal viewpoint?
Special viewpoint	Is the action from a special viewpoint?
Moving camera	Is the camera moving during the action?
Shotcut	Does the action extend over a shotcut?
Occlusion	Is the part of the video where the action is (spatially) located occluded at some time during the action?
Spatial truncation	Does the action extend beyond frame borders?
Start truncated	Is the start of the action missing?
End truncated	Is the end of the action missing?

Table 1: Specification of the provided metadata of the TVSeries dataset.

There is a large amount of variability in this dataset, making it more challenging than the most realistic datasets currently used. First, every actor does an action his or her way. Second, different actions can occur at the same time, being performed by the same or multiple actors. Third, the way the action is recorded can be very different: viewpoint, occlusion... Sometimes, the recording only starts after the action has started, or it ends too early. Some actions are not captured clearly, others are performed by bystanders in the background and are very small. Fourth, the camera can be moving and there are many shotcuts. Actions extend over multiple shots.

For every action instance, we provide metadata labels that give more information on how the action is performed and captured (see Table 1). This dataset will be made publicly available to encourage further research on (online) action detection on realistic data.

3 Evaluation protocol

In online action detection, a decision needs to be made at every frame, for every action: how likely is it that the action is going on in that frame, based on the information available up to that point? Therefore, it is logical to use the average precision over all frames as a metric for the performance of an online action detector. This metric has one big disadvantage,

[1] *Breaking Bad* (3 episodes), *How I Met Your Mother* (8), *Mad Men* (3), *Modern Family* (6), *Sons of Anarchy* (3) and *24* (4)

though: it is sensitive to changes in the ratio of positive frames versus negative background frames (if the classifiers are not perfect). If there is (relatively speaking) more background data, the probability increases that some background frames are falsely detected with higher confidence than some true positives: the AP will decrease. This makes it hard to compare the AP of two different classes when they do not have the same positive vs. negative ratio. Likewise, it makes it hard to evaluate performance on subsets of the data (*e.g.*, performance of unoccluded vs. occluded instances). To enable an easy, fair comparison, we introduce the *calibrated precision*, $cPrec = TP/(TP + \frac{FP}{w}) = w*TP/(w*TP+FP)$. We choose w equal to the ratio between negative frames and positive frames, such that the total weight of the negatives becomes equal to the total weight of the positives. Based on this calibrated precision, we can compute the *calibrated average precision (cAP)*, similar to the AP. This way, the average precision is calculated as if there were an equal amount of positive and negative frames: the random score is 50%. This evaluation metric is inspired by the work of Hoiem *et al.* [8] and Jeni *et al.* [10].

For our dataset, we take the mAP as final performance measure. To compare the effectiveness of the different classifiers and the influence of the metadata labels, we use the cAP instead.

4 Experiments

4.1 Baseline features

We analyze the difficulty of our dataset with three baseline methods, that are the backbone of most action detection systems today.

1. Trajectories + FV In our first approach, we calculate improved trajectories [17] and Fisher vectors [14] as in [17]. We train a linear SVM using fixed-length windows of 20, 40, 60 and 80 frames and use max-pooling to obtain a score for every frame.

2. CNN As a second approach, we run a CNN on every frame separately. We choose the VGG-16 architecture [16]. Since our training data is relatively small, we first pre-train our model on UCF101 split-1, then we finetune on our dataset. We also do image flipping and multiscale cropping for data augmentation. As CNN relies on single frames only, there is no temporal information encoded.

3. LSTM Our third approach is based on the recently successful LSTM [3, 18]. We use a single layer LSTM architecture with 512 hidden units. The fc6 features calculated with our CNN are then fed into the LSTM. For training and testing, each video is split into multiple sequences of 16 frames (stride 1). Our LSTM model takes 16 frames as input at a time, and returns class probabilities for the last frame.

4.2 Offline detection

In offline detection, the goal is to find the start and end frame of any action that occurs in the video. All information of the video is available at once, and calculation time is not an issue. As this is a more widely studied setting, we first report offline detection scores on our new dataset using the methods described above, as a reference.

To this end, the baselines need to be adapted to the offline setting. For baseline 1, we use a non-maximum suppression algorithm (as in [5]) to eliminate double detections. For baseline 2 and 3, we use windows with as length the median of the duration of the instances of that class. We then use the same non-maximum suppression algorithm.

Evaluation is done in the traditional setting, with intersection over union. We obtain a mAP for overlap ratio 0.2 of 4.9%, 1.1% and 2.7% for FV, CNN and LSTM respectively. These detection scores are quite low, indicating that this is a difficult dataset. For reference: the average classification accuracy of the actions (without taking the background into account), is 15.3%, 24.7% and 22.4% for FV, CNN and LSTM.

4.3 Online detection

In online detection, we decide at every moment whether a specific action is happening *now*. We evaluate by reporting the average precision over frames, as discussed in Section 3. The mAP is 5.2%, 1.9% and 2.7% for the FV, CNN and LSTM respectively. The values are very low, because the amount of negative data is very high, but still clearly better than the random mAP of 0.6%. Here too, FVs score higher than LSTM and CNN.

To be able to compare the scores of the different classes, we calculate the cAP. In general, FVs are better than LSTM, which is better than CNN. The three methods perform best on different classes. FVs capture motion information, and are therefore best for classes that inherently have a lot of motion, like 'run' and 'punch', as opposed to actions like 'write' and 'eat'. CNN on the other hand is appearance-based, and therefore needs characteristic poses or context information from objects and scenes. It works best for 'fire weapon' and 'get in/out car'. The AP is lower than the AP of the FVs: with realistic data, this static information is not sufficient. LSTM uses the CNN features and is able to use their temporal order. This is not the same as having real motion information, but a step in the right direction (reflected by its score in between CNN and FV).

5 Conclusion

Online action detection is a difficult problem, that has not been studied in a real-world setting and with realistic data before. We collected a new dataset and proposed an evaluation protocol to assist the research on online action detection. We tested a few baselines and showed none of the simple methods perform well. Online action detection is a novel problem far from being solved.

[1] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *CVPR*, 2016.

[2] Y. Cao, D. Barrett, A. Barbu, S. Narayanaswamy, H. Yu, A. Michaux, Y. Lin, S. Dickinson, J.M. Siskind, and S. Wang. Recognize human activities from partially observed videos. In *CVPR*, 2013.

[3] J. Donahue, L.A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[4] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Rank pooling for action recognition. *TPAMI*, 2016.

[5] A. Gaidon, Z. Harchaoui, and C. Schmid. Actom sequence models for efficient action detection. In *CVPR*, 2011.

[6] M. Hoai and F. De la Torre. Max-margin early event detectors. In *CVPR*, 2012.

[7] M. Hoai and F. De la Torre. Max-margin early event detectors. *IJCV*, 107(2):191–202, 2014.

[8] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. In *ECCV*. 2012.

[9] M. Jain, J. van Gemert, H. Jegou, P. Bouthemy, and C.G.M. Snoek. Action localization with tubelets from motion. In *CVPR*, 2014.

[10] L. Jeni, J. Cohn, and F. De La Torre. Facing imbalanced data–recommendations for the use of performance metrics. In *ACII*. IEEE, 2013.

[11] Y. Kong, D. Kit, and Y. Fu. A discriminative model with multiple temporal scales for action prediction. In *ECCV*. 2014.

[12] T. Lan, T.-C. Chen, and S. Savarese. A hierarchical representation for future action prediction. In *ECCV*.

[13] I. Laptev. On space-time interest points. *IJCV*, 2005.

[14] F. Perronnin, J. Sánchez, and T. Mensink. Improving the fisher kernel for large-scale image classification. In *ECCV*. 2010.

[15] M. Ryoo. Human activity prediction: Early recognition of ongoing activities from streaming videos. In *ICCV*, 2011.

[16] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015.

[17] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[18] S. Yeung, O. Russakovsky, N. Jin, M. Andriluka, G. Mori, and L. Fei-Fei. Every moment counts: Dense detailed labeling of actions in complex videos. *arXiv:1507.05738*, 2015.